EHB420E – Artificial Neural Networks Final Project

Özgür Karaaslan
0401090092

I designed a machine learning algorithm to predict orientation and landing position of a coin dropped from 1 meter. Also, I collected my own data for training the model.

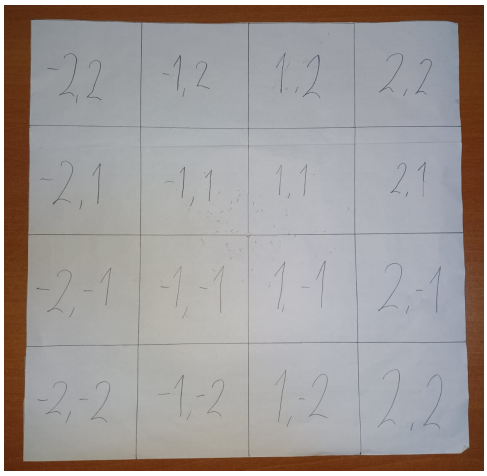
Data Labeling and Collection:

- I arranged various parameters related coin dropping and determined columns of data file.

  Columns:
  a) drop_number : Number of each coin dropping.
  b) drop_orientation : Orientation of coin before dropping. Values are 'H' and 'T'.
  c) drop_angle_surface : Angle of coin with surface. Values are 0, 37,70
  d) drop_polar_angle : Polar angle of coin. Values are 0, 45, 90, 135, 180, 225, 270, 225
  e) final_orientation : Orientation of coin in dropped situation.
  f) final_location_x : X coordinate of coin in dropped situation.
  g) final_location_y : Y coordinate of coin in dropped situation.

- I created csv file named "data.csv" and added these columns.

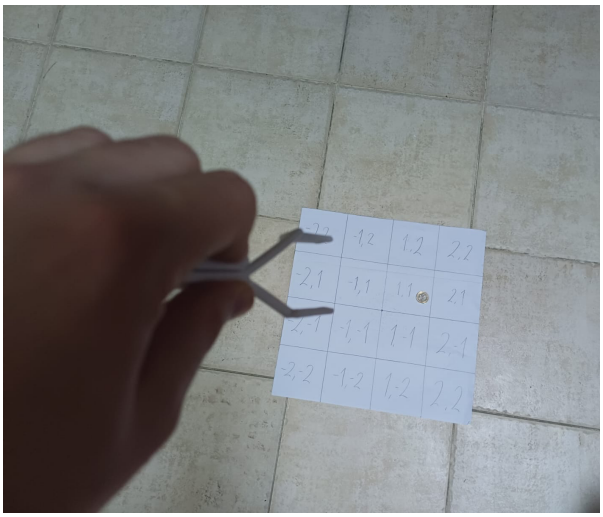
- I prepared an experiment setup for the data collection. At first, I create a grid for handling position of dropped coin. The coordinates consist x and y values. It can be seen in picture below. Also, I printed tweezers from my 3D printer to drop coin correctly. If I would drop coin with my hand, it cannot be appropriate dropping.



the Grid


the Tweezers

- After preparing experiment setup, I move forward to data collection part.
- I dropped coin 100 times according to dropping parameters above.
- I take note of dropping results and fill them to the data.csv file.



- “data.csv” file:

	A	B	C	D	E	F	G	H
1	drop_number,drop_orientation,drop_angle_surface,drop_polar_angle,final_orientation,final_location_x,final_location_y							
2	1,H,0,0,T,1,1							
3	2,H,35,45,H,2,1							
4	3,H,70,90,T,-1,1							
5	4,H,0,135,T,-1,1							
6	5,H,35,180,H,-1,-1							
7	6,H,70,225,H,-1,-1							
8	7,H,0,270,H,-1,-2							
9	8,H,35,315,T,2,1							
10	9,H,70,0,T,1,1							
11	10,H,0,45,T,-1,1							
12	11,H,35,90,H,-1,1							
13	12,H,70,135,H,1,-1							
14	13,H,0,180,H,1,2							
15	14,H,35,225,H,-1,1							
16	15,H,70,270,T,1,1							
17	16,H,0,315,H,1,-2							
18	17,H,35,0,T,-1,-1							
19	18,H,70,45,T,-1,-1							
20	19,H,0,90,H,-1,1							
21	20,H,35,135,T,-1,-1							
22	21,H,70,180,T,1,1							

Model Selection:

In my project, I chose to use Support Vector Machines (SVM), employing Support Vector Classifier (SVC) for classification and Support Vector Regressor (SVR) for regression tasks. This decision was based on SVM's remarkable ability to handle complex and high-dimensional data like mine. Its strength lies in identifying and predicting intricate patterns, which is essential for accurately classifying orientations and predicting precise locations in my dataset. One of the key reasons I leaned towards SVM is its capacity to strike a smart balance between precision and simplicity, ensuring that my model is not just accurate but also avoids unnecessary complexity. This combination of efficiency and practicality makes SVM an ideal choice for the nuanced requirements of my project.

I evaluated to use other machine learning methods but this methodology is more proper for my project. Moreover, using different approaches for location and orientation results affected the model positively.

Code Implementation:

I developed a program that predicts orientation and x-y coordinates using the dataset I prepared. I used the Python language for this task. I used the Pandas library to read the data from a .csv file. For implementing SVM algorithms and performing operations on the data (such as obtaining test and train data), I used the Scikit-Learn library. I performed hyperparameter tuning to enhance the predictive capability of the program.

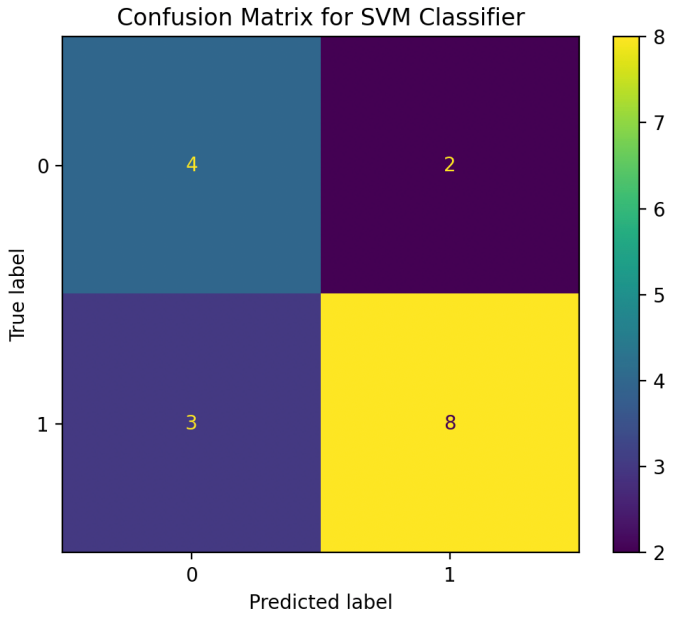
I also added explanations with code comments. The code and comments are available in this Github repository:

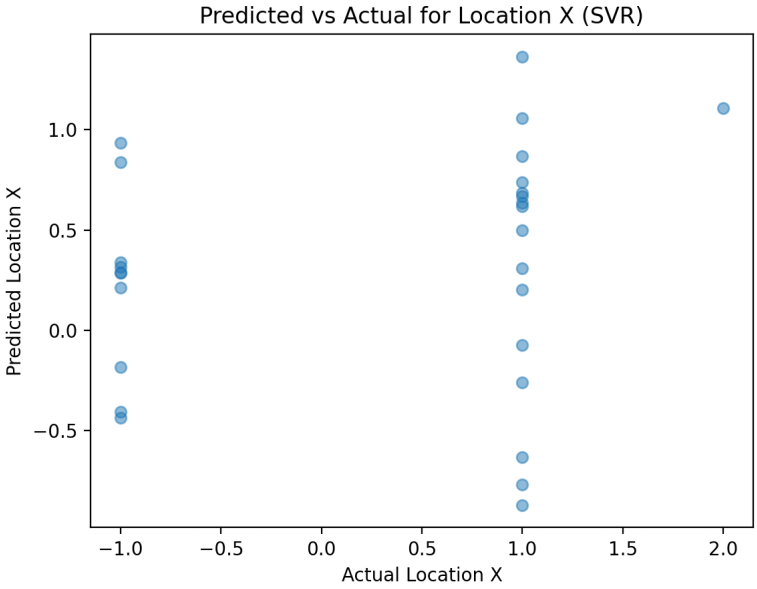
https://github.com/ozgurkaraaslan/Coin_drop_prediction

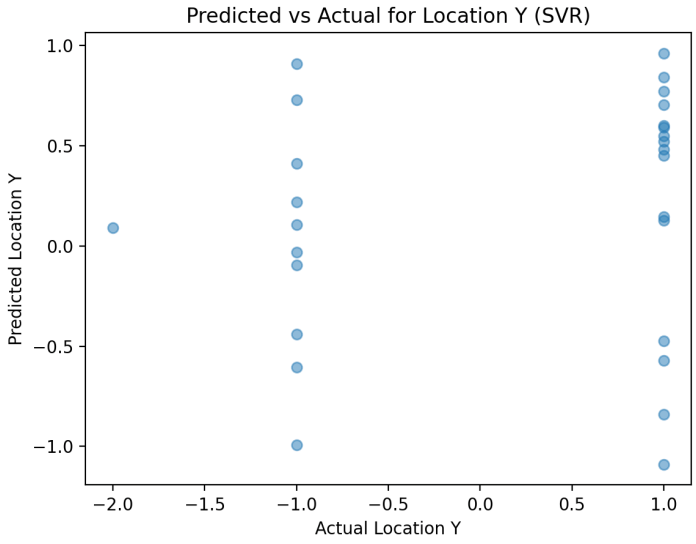
Code Outputs:

```
SVM Classifier Performance:
Accuracy for Orientation Prediction: 0.7058823529411765
```

```
MSE for Location X: 1.1723104746312292
MSE for Location Y: 1.2333313638581302
```







Detailed Classification Report for SVM Classifier:

	precision	recall	f1-score	support
0	0.57	0.67	0.62	6
1	0.80	0.73	0.76	11
accuracy			0.71	17
macro avg	0.69	0.70	0.69	17
weighted avg	0.72	0.71	0.71	17